第一章 操作系统引论.

一、名词解释.

1. 操作系统：管理计算机硬件与软件的软件系统.

4. 实时操作系统：指系统能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行.

5. 互斥共享：系统中的某些资源，如打印机，磁带机等，虽然可以提供给多个进程(线程)使用，但应规定在一段时间内只允许一个进程访问该资源.

三、填空题.

1. 硬件，软件.

2. 硬件资源，软件资源.

5. 共享性，异步性.

第二章 进程的描述与控制.

一、名词解释.

3. 临界区：每个进程中访问临界资源的代码段.

4. 进程同步：进程间相互合作.

三、填空题.

1. 动态性，异步性.

5. 动态，静态.

6. 进程同步

8. 共享存储器系统，管道通信系统，消息传递系统.

9. 执行，就绪，阻塞.

五、问答题.

2. $S_1 \to S_2$，$S_1 \to S_3$，$S_2 \to S_4$，$S_2 \to S_5$，$S_3 \to S_6$，$\{S_4, S_5, S_6\} \to S_7$

```
P1(){ S1; V(se1_2); V(se1_3);}        // P1()表示进程.
P2(){ P(se1_2); S2; V(se2_4); V(se2_5);}
P3(){ P(se1_3); S3; V(se3_6);}
P4(){ P(se2_4); S4; V(se4_7);}
P5(){ P(se2_5); S5; V(se5_7);}
P6(){ P(se3_6); S6; V(se6_7);}
P7(){ P(se4_7); P(se5_7); P(se6_7); S7;}.
void main(){
  semaphore se1_2, se1_3, se2_4, se2_5, se3_6, se4_7, se5_7, se6_7;
  se1_2.value = se1_3.value = se2_4.value = se2_5.value = 0;
  se3_6.value = se4_7.value = se5_7.value = se6_7.value = 0;
  cobegin
     P1(); P2(); P3(); P4(); P5(); P6(); P7();
  coend
}
```

3.

```
void father(){
  while(1){
    P(plate);
    P(mutex);
    放水果;
    V(mutex);
    if(是苹果) V(apple);
    if(是柑子) V(orange);
  }
}
```

```
void son(){
  while(1){
    P(orange);
    P(mutex);
    拿柑子;
    V(mutex);
    V(plate);
    吃柑子;
  }
}
```

```
void daughter(){
  while(1){
    P(apple);
    P(mutex);
    拿苹果;
    V(mutex);
    V(plate);
    吃苹果;
  }
}
```

```
void main(){ semaphore plate, orange, apple, mutex;
    plate.value = 5; mutex = 1;
    orange.value = apple.value = 0; cobegin father(); son(); daughter();
    coend}
```

4.

```
void PA(){
 while(1){
   P(me1); P(me3);
  将文件从磁盘读入缓冲1;
   V(me1); V(me3);
 }
}
```

```
void PB(){
 while(1){
   P(me2); P(me1);
  从缓冲1复制到缓冲区2;
   V(me1); V(me2);
 }
}
```

```
void PC(){
 while(1){
   P(me2); P(me4);
  将缓冲2内容打印;
   V(me2); V(me4);
 }
}
```

```
void main(){
  Semaphore me1, me2, me3, me4; // me1, me2: 缓冲区1、2. me3:磁盘. me4:打印机.
   me1.value = me2.value = me3.value = me4.value = 1;
   cobegin
     PA(); PB(); PC();
   coend
}
```

5.

```
void barber(){
   while(1){
    if(c==0) 睡觉;
    P(c); V(sofa);
    理发;
   }
}
```

```
void customer(){
    if(sofa==0) 离开;
    else {
      if(c==0){唤醒理发师; V(c);}
      else {P(sofa); 坐下等待;}
    }
}
```

```
void main(){
    Semaphore sofa=N, c=0;
    cobegin barber(); customer(); coend
}
```

第三章 处理机调度与死锁.

一、名词解释

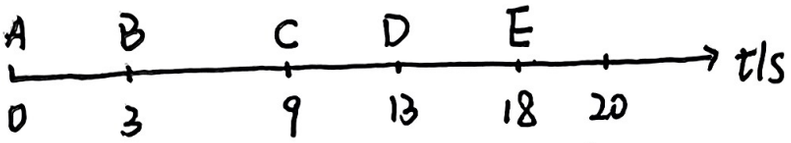
2. 处理机调度：对处理机资源进行分配，核心为优先级排序
3. 周转时间：从作业被提交给系统开始，到作业完成为止的这段时间间隔.
4. 死锁：多个进程因相互等待对方持有的资源而陷入的一种僵局状态.

三、填空题

2. 后备、完成、提交.
6. 预防、避免、检测、解除.

五、问答题.

1. ①先来先服务



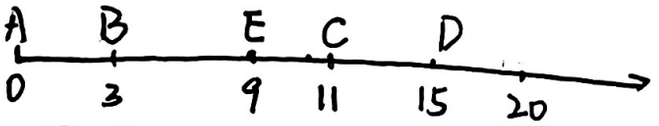
完成时间：A:3s. B:9s. C:13s. D:18s. E:20s.

周转时间：A:3-0=3s. B:9-2=7s. C:13-4=9s. D:18-6=12s. E:20-8=12s.

带权周转时间：A:$\frac{3}{3}=1$. B:$\frac{6}{7}$ C:$\frac{4}{9}$ D:$\frac{5}{12}$ E:$\frac{2}{12}=\frac{1}{6}$.

平均带权周转时间：$W=\frac{1}{5}(1+\frac{6}{7}+\frac{4}{9}+\frac{5}{12}+\frac{1}{6})=\frac{1}{5}\times\frac{691}{252}=\frac{691}{1260}$

②非抢占SJF：



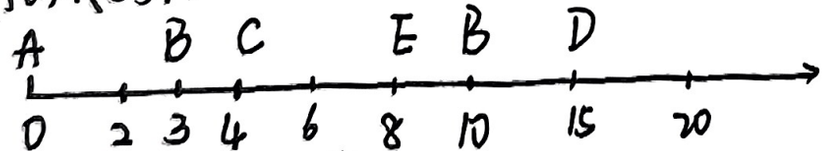
完成时间：A:3s. B:9s. E:11s. C:15s. D:20s.

周转时间：A:3-0=3s. B:9-2=7s. C:15-4=11s. D:20-6=14s. E:11-8=3s.

带权周转时间：A:$\frac{3}{3}=1$ B:$\frac{7}{7}$ C:$\frac{11}{11}$ D:$\frac{14}{14}$ E:$\frac{3}{3}$

平均带权周转时间：$W=\frac{1}{5}(1+\frac{7}{7}+\frac{11}{11}+\frac{14}{14}+\frac{3}{3})=\frac{1}{5}\times\frac{1601}{462}=\frac{1601}{2310}$.

③抢占式SJF.



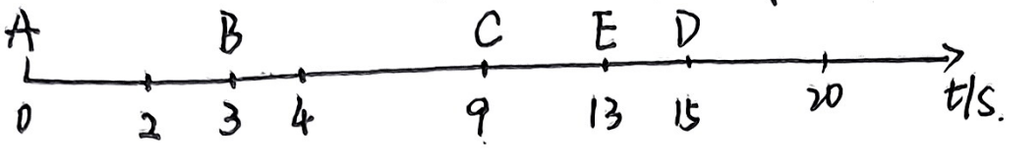
完成时间：A:3s. B:15s. C:8s. D:20s E:10s.

周转时间：A:3-0=3s. B:15-2=13s. C:8-4=4s. D:20-6=14s. E:10-8=2s.

带权周转时间：A:$\frac{3}{3}=1$. B:$\frac{13}{13}$ C:$\frac{4}{4}=1$. D:$\frac{14}{14}$ E:$\frac{2}{2}=1$.

平均带权周转时间：$W=\frac{1}{5}(1+\frac{13}{13}+1+\frac{14}{14}+1)=\frac{1}{5}\times\frac{695}{182}=\frac{139}{182}$

④ HRRN $R_p = \frac{响应时间}{要求服务时间}$ 非抢占式.



9s时: $R_{PC} = \frac{4+(9-4)}{4} = \frac{9}{4}$. $\frac{9}{4} > \frac{3}{2} > \frac{8}{5}$ C开始

$R_{PD} = \frac{5+(9-6)}{5} = \frac{8}{5}$

$R_{PE} = \frac{2+(9-8)}{2} = \frac{3}{2}$

13s时: $R_{PD} = \frac{5+(13-6)}{5} = \frac{12}{5} < R_{PE} = \frac{2+(13-8)}{2} = \frac{7}{2}$ E开始

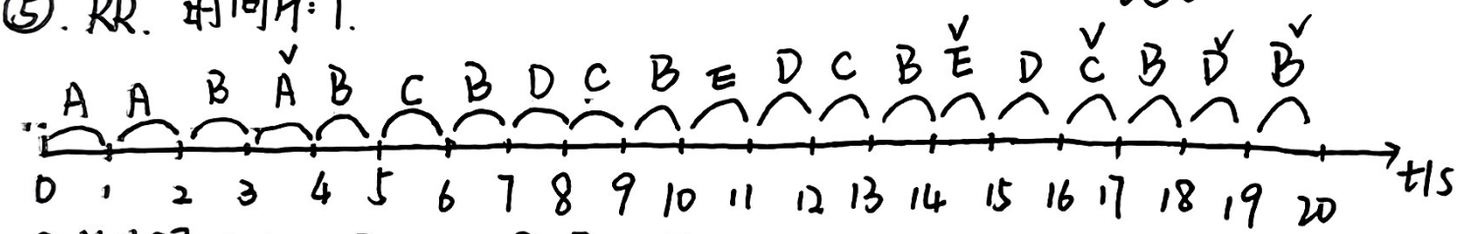
完成时间: A: 3s. B: 9s C: 13s. D: 20s. E: 15s.

周转时间: A: 3-0=3s. B: 9-2=7s. C: 13-4=9s. D: 20-6=14s. E: 15-8=7s.

带权周转时间: A: $\frac{3}{3}=1$. B: $\frac{7}{7}$ C: $\frac{9}{9}$ D: $\frac{14}{14}$ E: $\frac{7}{7}$

平均带权周转时间: $W = \frac{1}{5}(1+\frac{7}{7}+\frac{9}{9}+\frac{14}{14}+\frac{7}{7}) = \frac{1}{5}\times\frac{371}{126} = \frac{371}{630}$

⑤. RR. 时间片: 1.



完成时间: A: 4s. B: 20s. C: 17s. D: 19s. E: 15s.

周转时间: A: 4-0=4s. B: 20-2=18s. C: 17-4=13s. D: 19-6=13s. E: 15-9=6s.

带权周转时间: A: $\frac{4}{4}$ B: $\frac{18}{18}=\frac{1}{3}$ C: $\frac{4}{13}$ D: $\frac{5}{13}$ E: $\frac{2}{6}=\frac{1}{3}$.

平均带权周转时间: $W = \frac{1}{5}(\frac{4}{4}+\frac{1}{3}+\frac{4}{13}+\frac{5}{13}+\frac{1}{3}) = \frac{1}{5}\times\frac{329}{156} = \frac{329}{780}$

2. 假设从t=0s时开始所有任务同时到达.

t=0s: A: 20-0-10=10ms. B: 50-0-10=40ms C: 50-0-15=35ms.

$10 < 35 < 40$. A开始

t=10ms: A结束. B: 50-10-10=30ms. C: 50-10-15=25ms. $25 < 30$. C开始.

t=20ms: A: 40-20-10=10ms. B: 50-20-10=20ms. C: 50-20-5=25ms.

$10 < 20 < 25$. A开始

t=30ms: A结束. B: 50-30-10=10ms. C: 50-30-5=15ms. $10 < 15$. B开始

t=40ms: A=60-40-10=10ms. B:结束. C:50-40-5=5ms. 5<10. C开始

t=45ms: A=60-45-10=5ms. B结束. C:结束. A开始.

重复上述步骤.

3.

(1) ① Available > Need[0] Available += MAX[0]. = (1 6 6 6)

② Available > Need[3] Available += Max[3] = (1 15 14 10)

③ Available > Need[1] Available += Max[1] = (3 22 19 10)

④ Available > Need[2] Available += Max[2] = (6 28 29 20).

⑤ Available > Need[4]. Available += Max[4] = (6 34 35 30)

该计状态是安全的.

Allocation = Max - Need =

$$\begin{pmatrix} 0 & 0 & 3 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 3 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{pmatrix} \begin{matrix} \rightarrow P_0 \\ \rightarrow P_1 \\ \rightarrow P_2 \\ \rightarrow P_3 \\ \rightarrow P_4. \end{matrix}$$

(2). Request (1,2,2,2) ≤ Need[2] (2,3,5,6).

Request (1,2,2,2) ≤ Available (1,6,2,2).

假定可分配给 P_2. Available -= Request = (0,4,0,0).

Need[2] -= Request = (1,1,3,4).

进行安全性检查.可用资源 Available (0,4,0,0).已不能满足任何进程的需要.系统进入不安全状态.系统不分配资源.

(3). 是的.